ELSEVIER

Discrete Applied Mathematics 92 (1999) 193–203

DISCRETE APPLIED MATHEMATICS

states in DAWGs

Mathieu Raffinot *

Institut Gaspard Monge, Université de Marne-la-Vallée, Cité Descartes, Champs-sur-Marne, 77454 Marne-la-Vallée Cedex 2, France

Received 25 September 1996; revised 15 October 1998; accepted 26 October 1998

Abstract

Following the work of A. Blumer, A. Ehrenfeucht and D. Haussler, we obtain an asymptotic estimation of the average number of terminal states in the suffix directed acyclic word graph (DAWG – also called *suffix automaton*) under a Bernoulli model. We first extract an expression of the average from the structure of the DAWG. With a Mellin transform, we then obtain an asymptotic expansion of the form $\ln(n)/\ln(A) + C(A) + F(n)$ where n is the size of the word, A the alphabet size, $C(A)$ a function of A, and F an oscillating function with small amplitude. Finally, we compare theoretical results with experimental results.

Keywords: DAWG; Suffix automaton; Average case analysis; Finite automaton; Mellin transform

1. Introduction

Suffix directed acyclic word graphs (DAWGs) are powerful tools for textual pattern matching and lead to optimal algorithms in average, like the Backward Dawg Matching (BDM) for one pattern, or MultiBDM for several patterns (see [3]). Studies have been undertaken to calculate their sizes in terms of number of states and edges, so as to predict the maximal or average space needed by the algorithms which use them and to demonstrate some of their properties. Blumer et al. obtained in [1] an estimation of the average number of states and edges in DAWGs under a Bernoulli model. However, it is also of interest to estimate the average number of terminal states, principally (1) to understand the average behavior of algorithms which attach special action to terminal states of DAWGs, like BDM [1,4], (2) to have an idea of the average memory needed to construct the DAWG itself. The terminal states form a "suffix chain" which represents all the information needed to update a DAWG with a letter α (i.e. obtain DAWG($w\alpha$) from DAWG(w)) in the classical *on-line* algorithm of [1,4].

* *E-mail address:* raffinot@monge.univ-mlv.fr. (M. Raffinot)

0166-218X/99/$ - see front matter
PII: S0166-218X(99)00047-5

In this paper we give an asymptotic estimation of the number of terminal states of a DAWG under a Bernoulli model. The method used is adapted from [1]. We begin by obtaining an expression of the probability that one factor is representative of a terminal equivalent class. To analyze this function asymptotically, we use a complex transform, named Mellin's transform. Theoretical results are extracted using methods developed in algorithm analysis [6–8,11] during the last few years. These methods are clearer and simpler than the ones previously used in [1]. We obtain an asymptotic expansion of the form $\ln(n)/\ln(A) + C(A) + F(n)$ where n is the size of the word, A the alphabet size, $C(A)$ a function of A, and F an oscillating function with small amplitude. We then compare with experimental results on random word samples for different sizes of alphabet.

In [9], Jacquet and Szpankowski proved the same results as in [1] with a new approach, more precise but much more complicated, called *string-ruler approach*. It should be possible to obtain the same result as ours with their approach.

2. Notations and DAWGs

We denote by Σ the alphabet we are working with, and $A = |\Sigma|$ its size. A word x of length n is a finite sequence $x = x_1x_2 \dots x_n$ of letter(s) of Σ. The empty word ε is the unique word of length 0. A word $w \in \Sigma^*$ is a factor of $x \in \Sigma^*$ if x can be written $x = uwz$, with $u, v \in \Sigma^*$. We denote by $\mathrm{Fact}(x)$ the set of factor of x. A factor w of x is called a suffix if $x = uw$. The set of suffixes of x is called $\mathrm{Suff}(x)$.

DAWG (*also called suffix automaton*). A DAWG on a word $x = x_1x_2 \dots x_n$ is the minimal (incomplete) deterministic finite automaton that recognizes the set $\mathrm{Suff}(x)$. Given a factor w of the pattern x, *endpos*(w) is the set of all the pattern positions where an occurrence of w ends in x. Formally, given $w \in \mathrm{Fact}(x)$, we define $endpos(w) = \{i / \exists u, x_1x_2 \dots x_i = uw\}$. We call each such integer a *position*. For example, $endpos(baa) = \{3, 7\}$ in the word *baabbaa*. Notice that $endpos(\varepsilon)$ is the complete set of possible positions.

We define an equivalence relation $\equiv$ between factors of the pattern. For $u, v \in \mathrm{Fact}(x)$, we define

$$u \equiv v \quad \text{if and only if} \quad endpos(u) = endpos(v).$$

The nodes of the DAWG correspond to the equivalence classes of $\equiv$, i.e. to sets of positions. The DAWG of the word $x = baabbaa$ is given in Fig. 1.

A more complete presentation of DAWGs may be obtained in [2,3]. However, the previous definition is sufficient for our purpose.

3. Probabilistic analysis

We consider a probabilistic model of the Bernoulli type over Σ, i.e all letters are independent. We consider also that all letters have the same probability to appear, $1/A$. Let x be a word of length n over the alphabet Σ.

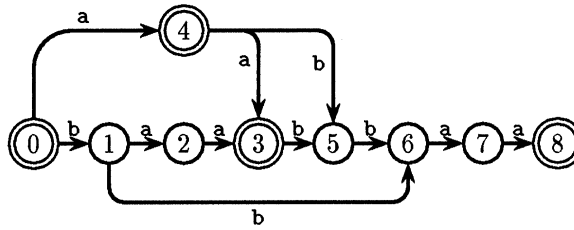

Fig. 1. DAWG of $x = baabbaa$. The double-circled nodes are terminals. The state 0 is the initial state.

A factor s of x is representative of its equivalence class if it is maximal in its class, i.e the longest word in its class. The properties verified by a DAWG insure any other word of this class to be a suffix of s.

We first express the average number of terminal states $F(n)$ as a sum on the word size n. This number is the probability for each word of length k, $0 \leqslant k \leqslant n$ to be representative of its equivalence class in DAWG(x). We separate in the count the prefixes of x in $G(n)$. We have

$$F(n) = G(n) + S(n),$$

with

$$G(n) = 2 + \sum_{k=1}^{n-1} A^k P_k^{(1)} \quad \text{and} \quad S(n) = \sum_{k=1}^{n-1} A^k P_k^{(2)}.$$

The term $P_k^{(1)}$ is the probability for a word of length $1 \leqslant k \leqslant n$ to be both a prefix and a suffix of x. The probability for a word of length k to be at a specific position in x is $1/A^k$ (we do not care about boundary effects, that become less significant as n grows). We have

$$P_k^{(1)} = \left(\frac{1}{A^k}\right)^2.$$

Since the entire word x is representative of its equivalence class and since this class will always be terminal, we add one node to the sum. Moreover, since the empty word ε is a suffix of x, the initial state of the DAWG of always terminal and we add it to the sum. This explains the isolated 2 in $G(n)$.

The term $P_k^{(2)}$ is the probability for a word of length k, which is not prefix of x, to be representative of a terminal class. This probability is expressed by a sum on the number of right contexts of the word

$$P_k^{(2)} = \sum_{m=2}^{n-1} P_{k,m},$$

where $P_{k,m}$ is the probability for a word of length k

- to appear exactly m times in x (at least two times, else it cannot represent an equivalence class);

- to be the longest which appears in its position (it means that it can not be extended on the left by the same letter at the m occurrences);
- to appear at a fixed position in x, in $n-k$.

The probability for a word of length k to appear in position $n-k$ is $1/A^k$. There are then $m-1$ occurrences to place in the word, which gives

$$\binom{n-2}{m-1}\left[\frac{1}{A^k}\right]^{(m-1)}\left[1-\frac{1}{A^k}\right]^{(n-1)-m}.$$

The probability for the same letter not to appear before all occurrences is $1-(A/A^m)$. We obtain for $P_{k,m}$:

$$P_{k,m}=\left[\left[\frac{1}{A^k}\right]\binom{n-2}{m-1}\left[\frac{1}{A^k}\right]^{(m-1)}\left[1-\frac{1}{A^k}\right]^{(n-1)-m}\right]\left[1-\frac{1}{A^{m-1}}\right].$$

We simplify S by expanding $P_k^{(2)}$:

$$P_k^{(2)}=\left[\frac{1}{A^k}\right]-\left[\frac{1}{A^k}\right]\left[1+\frac{1}{A^{k+1}}-\frac{1}{A^k}\right]^{n-2},$$

$$S(n)=\sum_{k=1}^{n-1}A^kP_k^{(2)}=\sum_{k=1}^{n-1}\left[1-\left(1+\frac{1}{A^{k+1}}-\frac{1}{A^k}\right)^{n-2}\right].$$

4. Approximations of S at infinity

We approximate the function S when $n\to\infty$. Let

$$S_0(n)=\sum_{k=1}^{n}\left[1-\left(1+\frac{1}{A^{k+1}}-\frac{1}{A^k}\right)^{n-1}\right].$$

We make in one step two approximations.

- First, by using the fact that $(1+x)^n\approx e^{nx}$ for a small x, we approximate $[1+(1/A^{k+1})-(1/A^k)]^{n-1}$ with $e^{(n-1)(1-A)/(A^{k+1})}$.
- We then replace the finite sum by an infinite one. We justify this approximation by the fact that the difference is growing and bounded independently of n, so that it converges to an $O(1)$ term.

The function to analyze becomes

$$T(x)=\sum_{k=1}^{\infty}1-e^{x\frac{(1-A)}{A^{k+1}}}$$

with $S(n+1)=S_0(n)\sim T(n-1)$.

Justification:

$$\begin{aligned}\mathrm{Diff}(n)&=|S_0(n)-T(n-1)|\\&<\sum_{k=1}^{n}\left[e^{(n-1)\frac{(1-A)}{A^{k+1}}}-\left[1+\frac{1}{A^{k+1}}-\frac{1}{A^k}\right]^{n-1}\right]+\sum_{k=n+1}^{\infty}1-e^{(n-1)\frac{(1-A)}{A^{k+1}}}.\end{aligned}$$

The first sum is bounded by a constant, independently of n, through an integral. For the second one, we use the inequality

$$0 < 1 - e^{-nx} < nx \quad \text{for} \quad n > 0 \text{ and } x > 0,$$

to obtain the bounds

$$0 < \sum_{k=n+1}^{\infty} 1 - e^{(n-1)\frac{(1-A)}{A^{k+1}}} < (A-1) \sum_{k=n+1}^{\infty} \frac{(n-1)}{A^k}.$$

This last sum is studied in [10], p. 131. It is bounded also from above by a constant independently of n. The total difference $\mathrm{Diff}(n)$ is then $O(1)$. Moreover, since $\mathrm{Diff}(n)$ is monotonous, the difference $\mathrm{Diff}(n)$ converges to a constant. We note this by $S_0(n) \sim T(n-1)$.

5. Asymptotic analysis of function T

We first present the asymptotic analysis tools (see [5–8,10,11]) we need to our purpose.

5.1. Asymptotic analysis tools

In order to obtain an asymptotic analysis of function T, we use a Mellin transform that map function on $[0, +\infty[$ to function on $\mathbb{C}$.

We need the Gamma function (Γ) on $\mathbb{C}$,

$$\Gamma(s) = \int_0^{\infty} e^{-t} t^{s-1} dt.$$

The integral converges for $s \in \mathbb{C}$ with $Re(s) > 0$.

The Γ function can be continued to all $s \in \mathbb{C}$, except at the poles $0, -1, -2, \ldots$. The residue of Γ at $-m$ is $(-1)^m/(m!)$.

Definition 1. The Mellin transform of a real valued function $F(x)$ locally Lebesgue integrable over $[0, \infty[$ is the complex function $F^*(s)$ of the complex variable s given by

$$F^*(s) = \int_0^{\infty} F(x) x^{s-1} dx.$$

Let $F(x)$ be piecewise continuous on $[0, \infty[$ and satisfy

$$F(x) = O(x^{\alpha}) \quad (x \to 0); \qquad F(x) = O(x^{\beta}) \quad (x \to \infty).$$

Then the Mellin transform is defined in the strip $-\alpha < Re(s) < -\beta$. This strip is called the *fundamental strip* of F^* and denoted by $\langle \alpha, \beta \rangle$.

Some classical Mellin transforms for exponentials are given in the following table.

$F(x)$	$F^*(s)$	Strip
e^{-x}	$\Gamma(s)$	$0 < Re(s)$
$e^{-x} - 1$	$\Gamma(s)$	$-1 < Re(s) < 0$
$e^{-x} - 1 + x$	$\Gamma(s)$	$-2 < Re(s) < -1$
$e^{x} - 1 - x - \frac{x^2}{2}$	$\Gamma(s)$	$-3 < Re(s) < -2$
$\log(1+x)$	$\frac{\pi}{s \sin \pi s}$	$-1 < Re(s) < 0$

Mellin transform has a very important functional property about *harmonics sums* [5,6]. Let $F(x)$ be

$$F(x) = \sum_{k=1}^{\infty} \lambda_k f(\mu_k x)$$

then its Mellin transform $F^*(s)$ is

$$F^*(s) = \left(\sum_{k=1}^{\infty} \lambda_k \mu_k^{-s} \right) f^*(s).$$

The use of Mellin transform stands on the fact (under general conditions) that it is possible to extract an asymptotic expansion of $F(x)$ from the poles of its transform $F^*(x)$. More precisely, the following theorem [6] makes the relationship.

Theorem 2 (P. Flajolet, X. Gourdon, P. Dumas). *Let $f(x)$ be continuous in $]0, +\infty[$ with Mellin transform $f^*(s)$ having a non-empty fundamental strip $\langle \alpha, \beta \rangle$. Assume that $f^*(s)$ admits a meromorphic continuation to $\langle \alpha, \gamma \rangle$ for some $\gamma > \beta$ and is analytic on $Re(s) = \gamma$. Assume also that there exists a real number $\eta \in (\alpha, \beta)$ such that*

$$f^*(s) = O(|s|^{-r}) r > 1$$

when $|s| \to \infty$ in $\eta \leqslant Re(s) \leqslant \gamma$. If $f^*(s)$ admits the singular expansion for $s \in \langle \eta, \gamma \rangle$

$$f^*(s) \asymp \sum_{(\xi,k) \in A} d_{\xi,k} \frac{1}{(s-\xi)^k},$$

then an asymptotic expansion of $f(x)$ in ∞ is

$$f(x) = - \sum_{(\xi,k) \in A} d_{\xi,k} \left(\frac{(-1)^{k-1}}{(k-1)!} x^{-\xi} (\ln(x))^{k-1} \right) + O(x^{-\gamma}).$$

5.2. Analysis

After computing the Mellin's transform $T^*(s)$ of the function $T(x)$, we obtain

$$T^*(s) = \frac{1}{(A-1)^s} \left[\frac{1}{A^s - 1} + \frac{1}{A^s} + 1 \right] \Gamma(s), \quad -1 < Re(s) < 0.$$

Poles of function $T^*(s)$ are

- One double pole at 0 given by the Γ function and the function $1/(A^s-1)$.
- Complex simple poles at $X_k = i\pi k/(\ln(A))$ for $k \in \mathbb{Z} \setminus \{0\}$ given by the function $1/(A^s-1)$.
- One pole at -1, but it is not relevant because we are interested in an asymptotic expansion when $x \to \infty$. According to the theorem, we may restrict our attention to the right part of the strip $\langle -1, 0\rangle$.

The double pole in 0 give as part of the asymptotic expansion of $T^*(s)$ the following terms in s^{-2} and s^{-1}:

$$\frac{1}{\ln(A)}s^{-2} + \left[\frac{3}{2} - \frac{\gamma + \ln(A-1)}{\ln(A)}\right] s^{-1}.$$

As 0 is the only real pole of $T^*(s)$, function $T(x)$ has for the principal part of its asymptotic expansion

$$T_0(x) = \frac{1}{\ln(A)}\ln(x) - \left[\frac{3}{2} - \frac{\gamma + \ln(A-1)}{\ln(A)}\right].$$

Complex poles add to this function an oscillating function with an exponentially increasing period. At $X_k = i2\pi k/\ln(A)$ for $k \in \mathbb{Z} \setminus \{0\}$, the residue of the expansion of

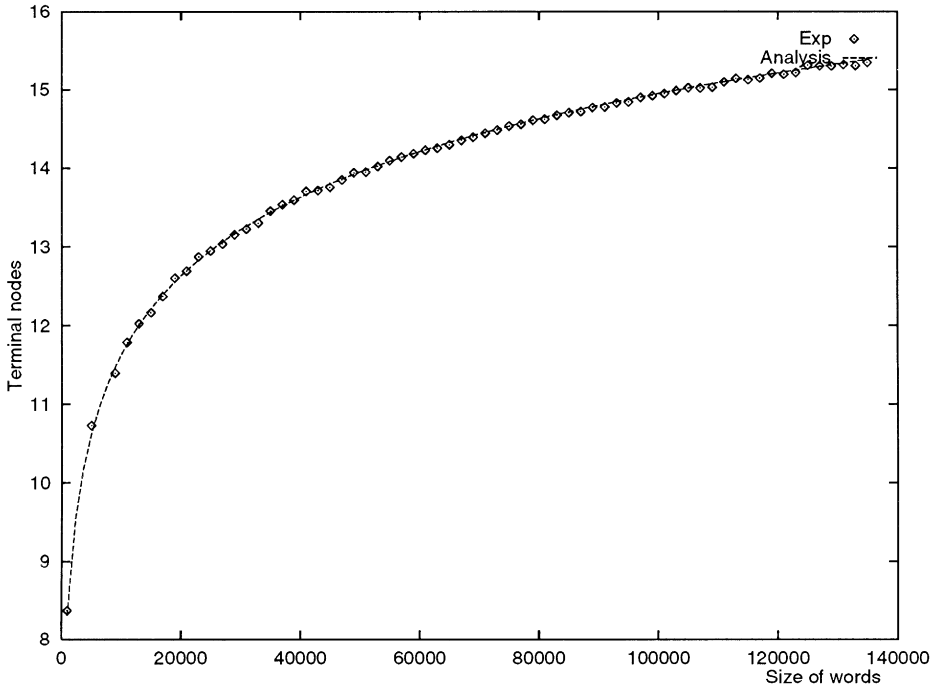

Fig. 2. Experimental results for $S(n)$ for an alphabet of size 2.

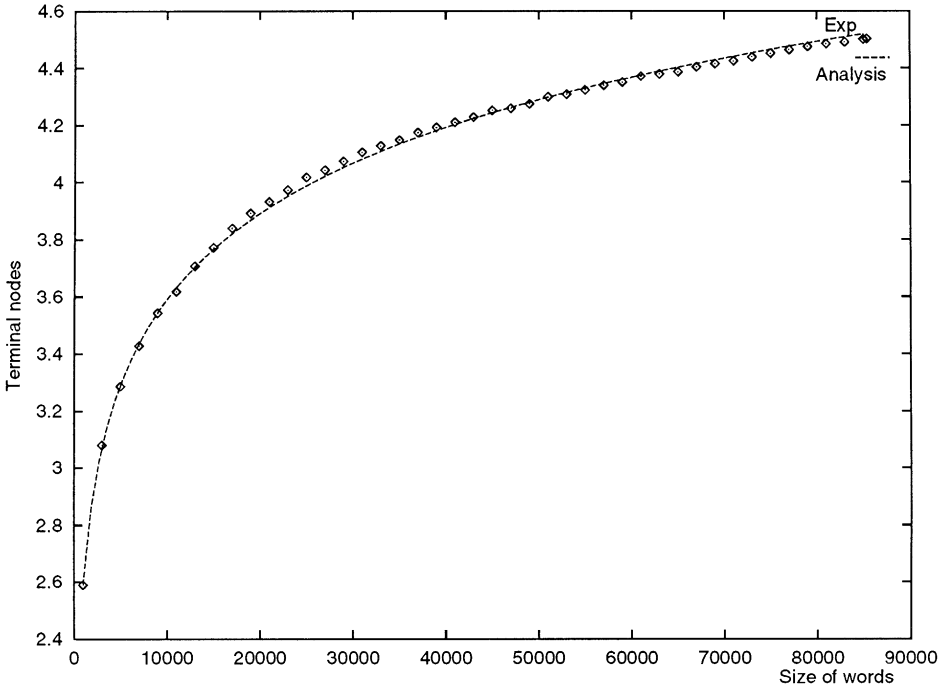

Fig. 3. Experimental results of $S(n)$ for an alphabet of size 10.

$1/(A^s - 1)$ is $1/\ln(A)$. Oscillating function $T_1(x)$ will be written as

$$T_1(x) = \frac{1}{\ln(A)} \sum_{k \in \mathbb{Z}\setminus\{0\}} \left[e^{-i\frac{2k\pi \ln(A-1)}{\ln(A)}} \Gamma(X_k) x^{-X_k} \right].$$

This can be also written as

$$T_1(\ln(x)) = \frac{1}{\ln(A)} \sum_{k \in \mathbb{Z}\setminus\{0\}} \left[e^{[-i\frac{2k\pi(\ln(x)+\ln(A-1))}{\ln(A)}]} \Gamma(X_k) \right].$$

Finally, we obtain for $T(x)$:

$$T(x) = C_1(A) + \frac{\ln(x)}{\ln(A)} + \frac{1}{\ln(A)} \sum_{k \in \mathbb{Z}\setminus\{0\}} \left[e^{[-i\frac{2k\pi(\ln(x)+\ln(A-1))}{\ln(A)}]} \Gamma(X_k) \right] + O(x^{-1}),$$

where $C_1(A)$ is a constant depending of A. Coming back to ours functions $S_0(n)$ and $S(n)$, we have

$$S(n) \sim C_2(A) + \frac{\ln(n-2)}{\ln(A)} + \frac{1}{\ln(A)} \sum_{k \in \mathbb{Z}\setminus\{0\}} \left[e^{[-i\frac{2k\pi(\ln(n-2)+\ln(A-1))}{\ln(A)}]} \Gamma(X_k) \right].$$

Analysis of $G(n)$. The function $G(n)$ can be approximated very easily by a constant when n grows.

$$G(n) = \frac{A}{A-1} - \frac{1}{A^{n-1}(A-1)} + 2 = \frac{A}{A-1} + 2 + O(n^{-1}).$$

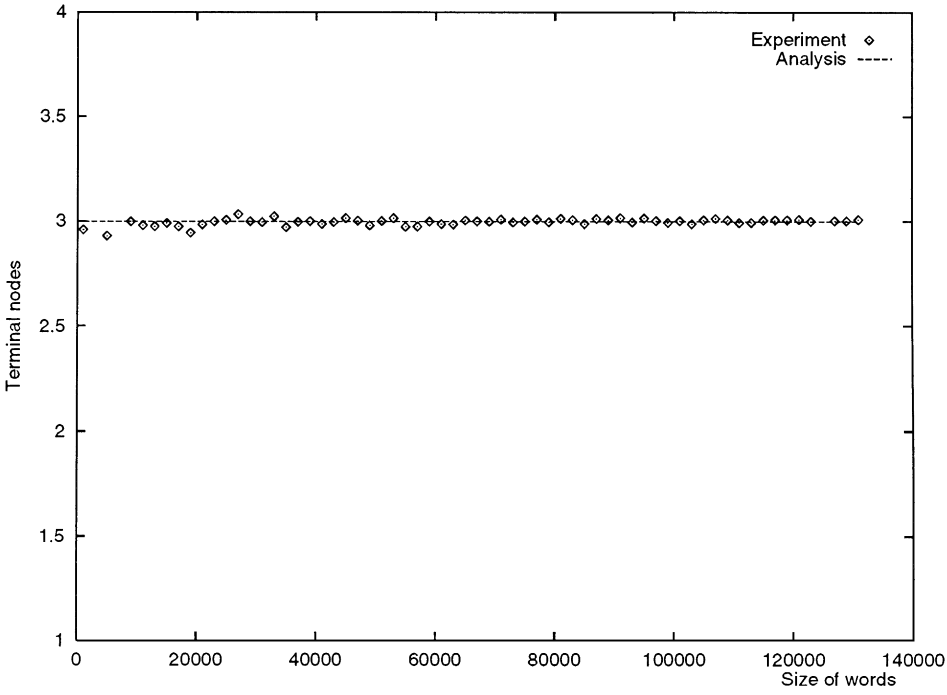

Fig. 4. Experimental results for $G(n)$ for an alphabet of size 2.

There is the final result for our initial function, the average number of terminals nodes, $F(n) = S(n) + G(n)$,

$$F(n) \sim C_3(A) + \frac{\ln(n-2)}{\ln(A)} + \frac{1}{\ln(A)} \sum_{k \in \mathbb{Z}\setminus\{0\}} \left[e^{[-i\frac{2k\pi(\ln(n-2)+\ln(A-1))}{\ln(A)}]} \Gamma(X_k) \right].$$

6. Experimental results

To compare theoretical and experimental results, we performed tests with two sizes of alphabet, $A = 2$ and $A = 10$, taking the average on a sample of words of length n with size varying from 1000 to 135 000 (using a step of 2000) for $A = 2$ and from 1000 to 85 000 (with the same step) for $A = 10$.

We first detail the part $S(n)$, then we turn to the part concerning $G(n)$.

6.1. Experimentals results for the function $S(n)$

It appears clearly in Fig. 2 that the experimental curve follows the main term $(\ln(n-1)/\ln(2))+C$ predicted by theory. Similarly, Fig. 3 shows the adequacy between theoretical results and the experimental ones obtained for $A=10$. We can also see that

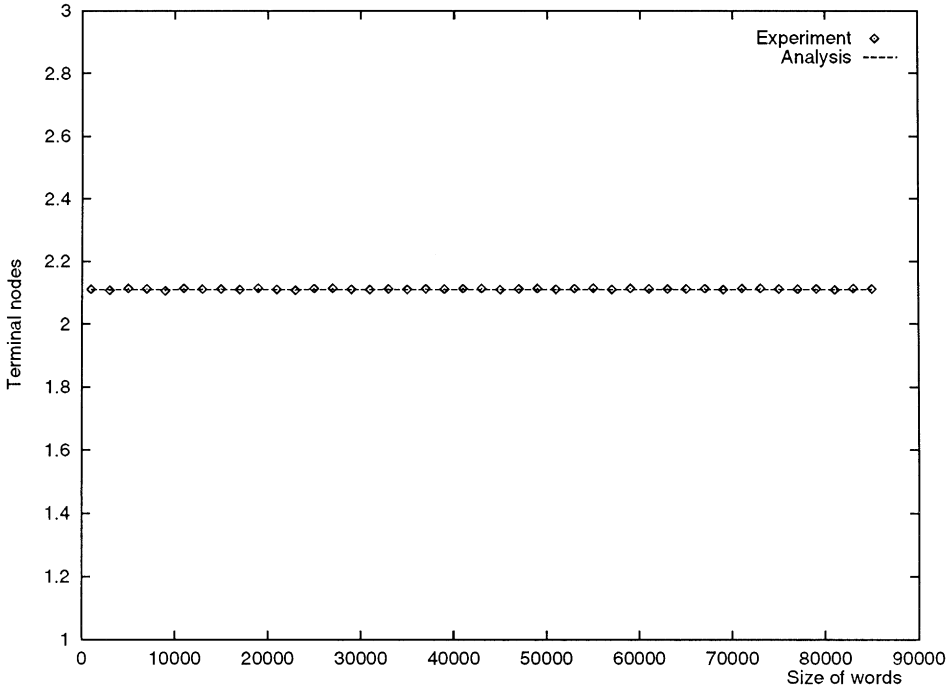

Fig. 5. Experimental results for $G(n)$ for an alphabet of size 10.

the oscillations, in respect to the main term in $\ln(n-1)/\ln(A)$, decrease very slowly to give in the limit the expected oscillations of order 10^{-6}. In both cases they stay in the order of 10^{-2}. This is due to the second order term given by the main theorem which is $O(1/x)$, and so very weak.

6.2. *Experimentals results for the function* $G(n)$

The comparison with theoretical results, for $|A| = 2$ in Fig. 4 or $|A| = 10$ in Fig. 5, shows the correctness of this analysis.

Computations have been done in parallel on six *HP9000* with the *PVM* library.

Acknowledgements

I would like to thank Philippe Flajolet and Maxime Crochemore for their help and precious advices. I would like also to thank Nicolas Bedon and Marie-France Sagot for checking this document.

References

[1] A. Blumer, A. Ehrenfeucht, D. Haussler, Average sizes of suffix trees and dawgs, Discrete Appl. Math. 24 (1) (1989) 37–45.
[2] M. Crochemore, C. Hancart, Automata for matching patterns, in: G. Rozenberg, A. Salomaa (Eds.), Handbook of Formal Languages, Springer, Berlin, 1996.
[3] M. Crochemore, W. Rytter, Text Algorithms, Oxford University Press, Oxford, 1994.
[4] A. Czumaj, M. Crochemore, L. Gąsieniec, S. Jarominek, Thierry Lecroq, W. Plandowski, W. Rytter, Speeding up two string-matching algorithms, Algorithmica 12 (1994) 247–267.
[5] P. Flajolet, X. Gourdon, P. Dumas, Mellin transforms and asymptotics: harmonic sums, Theoret. Comput. Sci. 144 (1–2) (1995) 3–58.
[6] P. Flajolet, M. Régnier, R. Sedgewick, Some uses of the Mellin integral transform in the analysis of algorithms, (Invited Lecture) in: A. Apostolico, Z. Galil (Eds.), Combinatorial Algorithms on Words, vol. 12 of NATO Advance Science Institute Series. Series F: Computer and Systems Sciences, Springer, Berlin, 1985, pp. 241–254.
[7] P. Flajolet, R. Sedgewick, The average case analysis of algorithms: complex asymptotics and generating functions, Research Report 2026, 1993, pp. 100.
[8] P. Flajolet, R. Sedgewick, The average case analysis of algorithms: counting and generating functions, Research Report 1888, 1993, pp. 116.
[9] P. Jacquet, W. Szpankowski, Autocorrelation on words and its applications, analysis of suffix tree by string-ruler approach, J. Combin. Theory Ser. A 66 (1994) 237–269.
[10] D. Knuth, The Art of Computer Programming 3: Sorting and Searching, Addison-Wesley, Reading, MA, 1973.
[11] R. Sedgewick, P. Flajolet, An Introduction to the Analysis of Algorithms, Addison-Wesley, Reading, MA, 1996, pp. 512.